AD 654162

Working Paper No. 117

A NEW APPROACH TO DISCRETE MATHEMATICAL PROGRAMMING

by

G. W. GRAVES AND A. B. WHINSTON

May, 1967

WESTERN MANAGEMENT SCIENCE INSTITUTE
University of California, Los Angeles

24

University of California

Los Angeles

Western Management Science Institute

Working Paper No. 117

"A NEW APPROACH TO DISCRETE MATHEMATICAL PROGRAMMING"

by

G. W. Graves and A. B. Whinston

May, 1967

This working paper should be regarded as preliminary in nature, and subject to change before publication in the open literature. It should not be quoted without prior consent of the authors. Comments are cordially invited.

This work was supported in part by contracts or grants from the Ford Foundation, National Science Foundation, and Office of Naval Research.

ABSTRACT

The paper presents an algorithm for solving the zero-one integer programming problem. In contrast to the usual approach, which reduces the problem to finding the optimal values for variables restricted to values of zero and one, we view the problem as finding an optimal map among a class of mappings of one finite set into another. The basic approach is to characterize the statistical structure of the finite class of maps. Our technique consists of trying to identify the optimal feasible map in a class of maps, by introducing random variables as functionals on the class of maps. We derive explicitly the statistical properties of the class of maps associated with the zero-one integer programming problem. Using the mean and variance-covariance matrix the idea of confidence level enumeration is developed.

A NEW APPROACH TO DISCRETE MATHEMATICAL PROGRAMMING

by

G. Graves and A. Whinston

Introduction

We shall present in this paper an algorithm for solving the following problem:

$$\text{Min. } \sum c_j x_j$$

$$(1) \qquad \text{s.t.} \quad \sum_j a_{ij} x_j \leq b_i \qquad i=1,2,\ldots,m$$

$$x_j \in I = \{0,1\}$$

The above linear integer programming problem is a specialization of the following:

$$\text{Min. } \sum_j f_j(x_j)$$

$$(2) \qquad \sum g_{ij}(x_j) \leq b_i \qquad i=1,2,\ldots,m$$

$$x_j \in I_j = \{0,1,2,\ldots\ldots s_j\}$$

Note that no conditions such as convexity or differentiability are imposed on the function. The algorithm to be outlined below can be used to solve problem (2). However, for purposes of exposition it is convenient to deal with problem (1).[1]

This paper presents the theoretical foundations for a new approach to integer programming. On the basis of the theoretical arguments, various specific computational procedures are developed which could be used to solve the integer programming problem. However, definitive computational results are not presented. In future papers we plan to explore the various computational options developed here.

The literature devoted to presenting methods for solving problem (1) is very extensive. A comprehensive survey has been presented by Balinski [2]. Roughly speaking the approaches to this problem can be divided into three groups. The first point of view initiated by Gomory involves generating the convex hull of the integer problem and then employing linear programming techniques. An elegant theory has been developed by Gomory to demonstrate, among other things, finite convergence. However, the computational experience with the method has proved inconclusive. This seems to be especially true for larger problems.[2]

The second approach can be characterized, in general, as an enumerative-combinatorial approach to the problem. Many researchers have made contributions to developing this theory. An early paper of Gilmore [5] presented an algorithm for the quadratic assignment problem within this framework. Recently work by Balas [1] and Glover [6] has developed an elegant enumerative algorithm to solve a linear integer problem. Basically the method involves an intelligent search through all possibilities. The ability to develop methods of excluding large numbers of possible solutions is the critical element in this approach. To this end bounds are developed in order to exclude solutions which would never be feasible or optimal. However, computational experience tends to suggest that because the bounds are not sufficiently informative at early stages of the computation, large problems cannot be handled. Roughly speaking problems where the number of variables exceeds 20 can not be effectively solved.

A third approach has been developed by Reiter and Sherman [10]. They have used statistical sampling techniques combined with methods for finding local optima. This approach appears to be quite effective for unconstrained problems. For the case where the problem has constraints, a large number of the samples may have to be rejected because of infeasibility.

The approach to be developed here might be considered an extension of the methods using enumeration. Instead of relying on lower bounds to truncate the process, we will introduce a new more powerful approach based on population statistics. The use of population statistics should not be confused with the sampling statistics or random search procedure described above.

Computational Scheme

This section will outline the computational scheme used in solving the problem. Typically, integer programming problems are considered from the point of view of finding the optimal integer values for a vector of variables. We shall view the problem in a different light - as one of selecting the optimal function among a certain class of functions which map elements of one set into another. For the type of problems discussed in this paper the class of functions is finite. A function or map is characterized by the way it assigns elements of one set to another set. By a k^{th} partial map we shall mean the subset of all functions which agree for some particular set of k elements of the domain of the maps. The maps in this class will differ in the way they assign to the remaining elements in the domain.

The integer programming problem can then be formulated as selecting from among the set of admissible maps the one which maximizes a linear functional. Note that not all maps would be admissible because of the constraints on the problem.

For the zero-one integer programming problem we may characterize the functions as those which map the integers $I = \{1,...n\}$ onto the set $\{0, 1\}$. There are 2^n such functions.

It is clear that one way of solving the problem is to enumerate all possible maps. As pointed out above this is obviously impractical. The first modification of a complete enumeration is to implicity enumerate all solutions. This simply means that in the course of our enumeration, certain subsets of maps can be ruled out without explicit examination.

There is sufficient information to show that these subsets could not contain a feasible optimal solution. These subsets are said to be implicitly enumerated.

In constructing an implicit enumeration scheme, there are four basic parts of the procedure;

1) A compact flexible enumerative scheme
2) Truncation by feasibility
3) Truncation by optimality
4) Selection of the next element to add to a partial specification of the mapping

Let us consider the idea of an enumerative scheme in the context of the above development. An enumerative scheme is a systematic way of generating all of the maps. It indicates which maps have been already generated and, of course, which ones remain to be enumerated. One enumerative procedure would be to simply list all maps that have been examined. However, for any reasonably sized problem, the number of possible maps would be too large. Simple listing would be impossible because of the enormous memory requirement. Thus a good enumerative scheme must allow for a compact representation of the required information. In addition to compactness, flexibility in the choice of successive maps is extremely valuable. For example, consider the following possible way of recording maps for the present problem. Start with the zero vector and add the number 1 successively in binary form until all binary vectors have been generated. Since a map can be uniquely represented by an n-vector of zeros and ones, this would enumerate all maps in a compact manner while indicating the number of solutions already tried. However, this scheme, while extremely compact, does not allow any flexibility of choice.

Within the class of enumerative schemes which allow for a compact storage, one should strive for a scheme which allows for maximal freedom in the sequence of successive maps generated. We thus present a general, flexible compact scheme for generating all admissible mappings of the set S into R. While below we present the procedure in a rather abstract manner the ideas underlying it are quite natural. One of the earliest uses of such ideas is due to Tarry [11] in his algorithm for finding a path through all vertices of a rooted graph. We proceed in the following manner where the steps follow consecutively, unless indicated:

0. Initialize by setting $k=1$ and determine S_1 and R_1 according to a specified rule, perhaps determined by feasibility.
1. If R_k is empty go to 7
2. Select any elements

$$i^* \in S_k$$
$$j^* \in R_k$$

Set $i_k=i^*$ and $j_k=j^*$ and $R_k=R_k-[j^*]$

3. If $k=n$ go to 11
4. Set $k+1=k$

5. Determine S_k and R_k according to some specified rule, perhaps determined by feasibility

6. Go to 1

7. If $k=1$ stop

8. Set $k-1=k$

9. If R_k is empty go to 7

10. Select any element $j^* \in R_k$. Set $j_k=j^*$ to go with the current i_k and set $R_k=R_k-j^*$ and go to 3

11. Record the current mapping

12. Go to 9

In the present case we take $S=S_1 = \{1,2,\ldots n\}$, $R=R_1 = \{0,1\}$ and in Step 5 we set

$$S_k=S-\bigcup_{t=1}^{k-1} i_t$$

and

$$R_k=\{0,1\}$$

These choices generate the desired maps. With the above specification the general enumerative scheme given here reduces to one proposed by Glover [6] to solve the 0-1 integer programming problem.

Another application of the general enumerative scheme is to the generation of all n! permutations of the integers $I=\{1,2,\ldots n\}$. For this application we set $S=S_1 = \{1,2,\ldots n\}$, $R=R_1 = \{1,2,\ldots n\}$ and in Step 5

$$S_k=S-\bigcup_{t=1}^{k-1} i_t$$

$$R_k=R-\bigcup_{t=1}^{k-1} j_t$$

A variant of this mapping scheme was used to solve the quadratic assignment problem as given in [8].

Step two in the enumerative scheme involves the selection of a particular variable, i.e.; an element from the set $S-\bigcup_{t=1}^{k-1} i_t$ and its assignment to either the element zero or one. Because of the powerful selection mechanism to be developed below, the flexibility of being able to select the variable is not needed in the present application. The order in which we treat the variables will be fixed. The enumerative procedure is altered slightly to account for this change. Let $(i_1,..i_n)$ be an ordering of the integers $(1,2,...n)$. Then set $S_1=(i_1)$, $S_2=(i_2)$ .. $S_n=(i_n)$. By renumbering the variables we may assume that $S_1=(1)$, $S_2=(2),...S_n=(n)$.

The enumerative procedure has the effect of transforming the integer programming problem into an n-stage decision problem. At the k^{th} stage of the algorithm we have to decide what value to assign the variable x_k. There are two considerations. First the local or immediate effect of setting x_k to a value. By setting $x_k=1$ the criterion value is increased by c_k (our immediate cost) and each constraint is altered by the value a_{ik}. The alternative choices of zero or one for x_k determine alternative subsequent problems in which possible choices for $x_{k+1}...x_n$ are differently restricted because of the a_{ik}. This second factor, the potential future cost, is the crucial factor. What is the cost of the alternative restrictions on the subsequent possible assignment of values for the remaining $x_{k+1}...x_n$?

If one were able to evaluate the second factors exactly it would be possible to assign at each step the correct value and solve the problem exactly in n steps. This is, of course, not possible. The next best thing to knowing exactly the values of the completions is to know them almost surely. The use of probability usually enables us to obtain very good information at a fraction of the cost of obtaining exact information. It is to capitalize on this that we develop the present approach. It should also be observed that in the large scheduling and allocation problems for which this algorithm is intended, the uncertainties in the underlying data reduce attempts to obtain the exact optimal solution with certainty to pedantry.

What is required for present purposes is the probability distribution of the values of the linear functionals over the discrete sample space of the remaining 2^{n-k} completions of the current k-partial map. Although it is impossible to obtain the exact multivariate distribution desired without complete enumeration, the really useful fact is that we can obtain the vector of means and covariance matrix of this distribution exactly and an asymptotic approximation to the distribution.

Statistical Properties of the Problem

Let x_j be random variables which have the following probability distributions:

$$Pr(x_j=1) = 1/2 \quad \text{and}$$

$$Pr(x_j=0) = 1/2$$

and further assume the x_j to be independent.

Then $z_i = \sum_{j=1}^{n} a_{ij}x_j$ is also a random variable where (a_{ij}) is any set of real constants. The means variances, and covariances of linear functions of random variables are well known.

$$\mu(z_i) = \sum_j a_{ij}\mu_j$$
$$\sigma^2(z_i) = \sum_j a_{ij}^2\sigma_j^2$$

and,

$$\| c(z_p, z_q) \| = \| \sum_j \sigma_j^2 a_{pj} \cdot a_{qj} \|,$$

where

$$\mu_j = E(x_j), \qquad \sigma_j^2 = E(x_j - \mu_j)^2 \quad \text{and}$$

$$c(z_p, z_q) = E[(z_p - \mu(z_p))(z_q - \mu(z_q))]$$

Now in our particular case since $\mu_j = 1/2$ and $\sigma_j^2 = 1/4$ these formulas reduce to:

$$\mu(z_i) = 1/2 \sum_j a_{ij}$$
$$\sigma^2(z_i) = 1/4 \sum_j a_{ij}^2$$

and,

$$\| c(z_p, z_q) \| = 1/4 \| \sum_j a_{pj} \cdot a_{qj} \|.$$

Further, the z_i are sums of independent random variables. Taking $x_{ik} = a_{ik}x_k$ then $E(x_{ik}) = a_{ik/2}$ and $\sigma^2(x_{ik}) = a^2_{ik/4}$ and $z_{in} = \sum_{k=1}^{n} x_{ik}$. The Lindeberg theorem [3] states: "Let $\tilde{x}_{i1}$, $\tilde{x}_{i2}, \ldots$ be mutually independent one-dimensional random variables with distributions $F_1, F_2 \ldots$. Assume $E(\tilde{x}_{ik}) = 0$ [e.g $\tilde{x}_{ik} = x_{ik} - E(x_{ik})$] and put $\sigma^2(z_{in}) = \sigma^2(x_{i1}) + \sigma^2(x_{i2}), \ldots + \sigma^2(x_{in})$.

Assume that for each $t > 0$

$$\lim_{n \to \infty} \frac{1}{\sigma^2(z_{in})} \sum_{k=1}^{n} \int_{|y| \geq t\,\sigma^2(z_{in})} y^2 \, F_k(dy) = 0$$

then the distribution of the normalized sum

$$\tilde{z}_{in} = \frac{z_{in} - E(z_{in})}{\sigma^2(z_{in})}$$

tends to the normal distribution N (0,1)."

Now assuming all the a_{ik} are from a bounded set, since $\sigma^2(z_{in}) \to \infty$ as $n \to \infty$ for any $t > 0$ the integrals $\int_{|y| \geq t} \frac{y^2 F_k(dy)}{\sigma^2(z_{in})} = 0$ for sufficiently large n, and the required condition for the Lindeberg theorem is satisfied. The fact that for large numbers of variables, $\tilde{z}_i = \frac{z_i - E(z_i)}{\sigma(z_i)}$ is approximately normal, enables one to estimate the minimum values of the z_i almost surely as follows: $P_r(\tilde{z}_i \leq c_\alpha) = \alpha$ is readily obtainable from normal tables for any confidence level α. Hence, $P_r(z_i \leq E(z_i) + c_\alpha \sigma(z_i)) = \alpha$ is available for each of the z_i.

The above results can be extended to obtain confidence level statements about the joint or coupled behavior of the z_i. The Lindeberg Theorem is readily extended to show that the vector $(\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_n)$ is asymptotically multivariate normal.[4]

Confidence Level Implicit Enumeration

In the course of the underlying enumeration it is necessary to apply the results of the previous section to the truncated random variables dictated by the successive fixing of elements of the map. To this purpose, we introduce the following definitions:

$$z_i(k) = a_{i,k+1} x_{k+1} + a_{i,k+2} x_{k+2} + \cdots + a_{in} x_n$$

for $1 \leq k < n$ and $z_i(n) = 0$,

$$\hat{z}_i(k) = a_{i1}\bar{x}_1 + a_{i2}\bar{x}_2 + \ldots a_{i,k-1}\bar{x}_{k-1}$$

for $1 < k \leq n$ and $\hat{z}_i(1) = 0$

where $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_{k-1})$ are the already chosen fixed values for these variables.

The criterion function is designated $z_c = c_1x_1+c_2x_2+\ldots+c_nx_n$. As is customary in the zero-one problem, we assume that the $c_j \geq 0$. This does not restrict complete generality inasmuch as the simple translation of variables $x_j' = 1-x_j$ for the $c_j < 0$ will insure this condition.

The most elementary use of the statistical properties developed in the previous sections is to supplement the bounds employed in the deterministic implicit enumeration as expounded by Balas, Glover and others. To see how this can be done, let us recapitulate the use of bounds for implicitly enumerating. Take $z_i^-(k) = \sum_{j \in S_i^-} a_{ij}$ where $S_i^- = \{j \mid a_{ij} < 0 \text{ and } k < j \leq n\}$ and clearly $z_i(k) \geq z_i^-(k)$. Now for a feasible solution

$$z_i = \hat{z}_i(k) + a_{ik}x_k + z_i(k) \leq b_i$$

or

$$z_i(k) \leq b_i - \hat{z}_i(k) - a_{ik}x_k = V_i(k)$$

So the possible completions of the partial map are implicitly enumerated if for neither value of x_k is

$$z_i^-(k) \leq b_i - \hat{z}_i(k) - a_{ik}x_k$$

while the value of x_k is fixed if the inequality holds for one value but not the other. The above argument applies as well to the criterion function when b_i is replaced by z_c^* the best known feasible value, with the additional fact that $S_c^- = \emptyset$ and hence $z_c^-(k)=0$.

The simplest use of the statistical properties of our problem would then be to use the estimated minimum

$$Z_i'(k) = E(Z_i(k)) + c_\alpha \sigma(Z_i(k)$$

in place of the absolute bound $\bar{Z}_i(k)$ in the above argument. The possible completions of the partial map are then implicitly enumerated with a confidence level of α. Confidence level implicit enumeration is an agreement not to check further those subsets of maps that have less than α% chance of containing a better feasible solution. It should be noted that by its construction $Z_i(k)$ is a random variable on the finite sample space of the remaining 2^{n-k} completion maps with equal probability of $\frac{1}{2^{n-k}}$ for each map. Now $V_i(k)=b_i-\hat{Z}_i(k)-a_{ik}x_k$ establishes a cutoff value (with a choice of a value for x_k) for a feasible solution of the i^{th} constraint. By fixing α and calculating the corresponding critical value $Z_i'(k)$ from the normal distribution if $Z_i'(k) > V_i(k)$, we can then infer the probability (or relative frequency) of the maps that would yield feasible solutions to the i^{th} constraint is less than α. By considering all of the constraints jointly, the probability that a feasible map could be found would be diminished. Schematically the situation would appear as follows(see fig. 1):

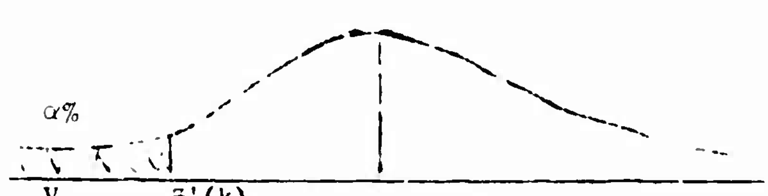

Fig. 1

The total shaded area contains α% of the maps and the area to the left of $V_i(k)$ contains less than α% of the maps. It should be observed that the truncation of the tails of the normal distribution because of our finite space approximation would cause the variances to be over estimated and hence make the procedure even more conservative. In fact as k approaches n and the normal approximation becomes less valid the procedure will undoubtedly automatically merge into the deterministic one where only the absolute bounds will cut.

A more advanced use of the statistical properties of the problem, which would undoubtedly yield much greater cutting power, would be to implicitly enumerate subsets of maps which have low percentages of maps which satisfy the constraints pairwise. Relying on the fact that any pair of the random variables $Z_i(k)$ and $Z_j(k)$ has an asymptotic bi-variate normal distribution, we can use the conditional probability distribution of $Z_j(k)$ given $Z_i(k)$. In this instance $Z_j(k)$ is asymptotically normally distributed about

$$\mu_j(Z_i(k)) = E(Z_j(k)) + \frac{c(Z_i(k), Z_j(k))}{\sigma^2(Z_i(k))}\left(Z_i(k) - E(Z_i(k))\right) \quad \text{with}$$

$$\text{variance} \quad \sigma^2_{j.i} = \sigma^2(Z_j(k)) \cdot \left(1 - \frac{c(Z_i(k), Z_j(k))}{\sigma(Z_i(k))\sigma(Z_j(k))}\right)^2 .$$

The conditional distribution of $Z_j(k)$ depends on the particular value of $Z_i(k)$ chosen. For the purpose of obtaining a critical point we may choose a realizable value of $Z_j(k)$ which gives the lowest and, therefore, weakest critical value. Now for $c(Z_i(k), Z_j(k)) < 0$ the minimum value of $\mu_j(Z_i(k))$ occurs by using

$$\gamma(i) = \min\left(z_i^+(k), V_i\right) \text{ for } z_i(k) \text{ where}$$

$z_i^+(k) = \sum_{p \in S_i^+} a_{ik}$ and $S_i^+ = \{p \mid a_{ip} > 0 \text{ and } k < p \leq n\}$.

With $c(z_i(k), z_j(k)) > 0$ the minimum value of $\mu(z_i(k))$ occurs by using $\gamma(i) = z_i^-(k)$.

In either instance one can use $z_j''(k;i) = \mu(\gamma(i)) + c_\alpha \sigma_{j.i}$ in place of $z_j'(k)$ in the preceding arguments. For each variable $z_i(k)$, $i \neq j$, a critical point $z_j''(k,i)$ is obtained. Let $z_j''(k)$ denote the maximum value. This value constitutes the best bound obtainable using the constraints pairwise. In practice, of course, the critical points would be generated sequentially. At each step, the algorithm checks whether the critical point exceeds $V_i(k)$ or not.

The most general use of the statistical properties of the problem would be to calculate directly the probability of the set of completion maps which simultaneously satisfy all the constraints and yield a better value of the criterion function than currently known. This would be,

$$\Pr\{z_c(k) \leq V_c(k), z_1(k) \leq V_1(k), \ldots, z_m(k) \leq V_m(k)\}$$

Theoretically, for each choice of a value for x_k, this probability can be calculated approximately from the asymptotic multivariate normal distribution. In practice although considerable research has been done on the subject over a large period of time, there does not appear to be effective means of computing directly this probability beyond the bi-variate case.[5]

In terms of the overall algorithm, information from the above tests is used in step 5. If, when setting $x_k=1$, we determine by use of the bounds that a possible completion is to be ruled out we set $R_k=R_k-\{1\}$. Similarly, in the case where $x_k=0$ is tested, we set $R_k=R_k-\{0\}$. If we could compute the multivariate normal then we rule out either or both values if their respective probabilities are less than a confidence level α.

Consider the case where neither value can be ruled out. Step two in the enumerative scheme calls for mapping the element $i^*=k$ into an element $j^* \{0,1\}=R_k$. This, of course, corresponds to selecting a value for x_k of zero or one. Note that either choice should lead to an improved feasible solution. While the enumerative scheme guarantees that if necessary both values will be checked, it is important for computational efficiency to make a judicious choice. By making a choice which leads to a feasible solution with a lower value, we obtain a sharper cut-off point. This cut-off point allows, in subsequent generation of maps, the implicit enumeration of larger subsets of maps than would be possible if a feasible, but higher value of the criterion function is obtained. In order to achieve this we elect to check first that subset of maps which contains a larger proportion of improved feasible solutions.

Conclusion

As is well known, many problems in combinatorial programming can be reduced to problems in variables restricted to the values zero and one by various artifices. The attempt to convert these problems to integer mathematical programming problems by introducing large numbers of structural constraints on the variables has not proved computationally effective. Combinatorial programs, almost by definition are associated with maps of one finite set into another and thus generate a finite class of maps which can be enumerated by the general algorithm presented here. It is our intention to attack these different problems by characterizing the statistical structure of the associated finite class of maps. Our technique consists of trying to identify the optimal feasible map in a class of maps by introducing random variables as functionals on the class of maps.

In an earlier paper [8] the authors have applied the above ideas to a different type of mapping scheme. Some preliminary computational results were given in that paper. Future papers will develop other types of maps and their statistical properties and show how these maps can be used to solve important industrial problems.

FOOTNOTES

1. For a discussion of methods of transforming problem (2) into the form of problem (1) see [1].

2. See e.g. [5] for a presentation of Gomory's approach to the problem.

3. See also [4].

4. See [3] for a discussion of such extensions.

5. In [9] a comprehensive survey of the area is given.

REFERENCES

[1] Balas, E., "Extension de l'algorithme additif à la programmation en nombres entiers et à la programmation non linèaire," Compte Rendus Acadamie Science, Vol. 258, pp. 5136-5139.

[2] Balinski, M. L., "Integer Programming: Methods, Uses, Computation," Management Science, Vol. 12, No. 3, pp. 253-313.

[3] Feller, W., An Introduction to Probability Theory and Its Application, Vol. II, John Wiley and Sons, 1966.

[4] Geoffrion, A. M., "Integer Programming by Implicit Enumeration and Balas' Method," forthcoming in S.I.A.M. Review.

[5] Gilmore, P. C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," Journal of the Society for Industrial and Applied Mathematics, Vol. 10, No. 2, pp. 305-313.

[6] Glover, F., "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem," Operations Research, Vol. 13, No. 6, pp. 879-919.

[7] Gomory, R. E., "Outline of an Algorithm for Integer Solutions to Linear Programs," Bulletin of the American Mathematical Society, Vol. 64 (1958), pp. 275-278.

[8] Graves, G. and A. Whinston, "An Algorithm for the Quadratic Assignment Problem," Working Paper No. 110, Western Management Science Institute.

[9] Gupta, S. S., "Probability Integrals of Multivariate Normal and Multivariate t," Annals of Mathematical Statistics, Vol. 34, No. 3, pp. 792-829.

[10] Reiter, S. and G. R. Sherman, "Discrete Optimizing," Journal of the Society for Industrial and Applied Mathematics, Vol. 13, No. 3, (1965), pp. 864-889.

[11] Tarry, G., "Le Problème des Labyrinthes," Nouvelles Annales de Math, Vol. 14, 1895, p. 187.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY *(Corporate author)* Western Management Science Institute University of California at Los Angeles	2a REPORT SECURITY CLASSIFICATION Unclassified 2b GROUP

3 REPORT TITLE

A New Approach to Discrete Mathematical Programming

4 DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Working Paper

5 AUTHOR(S) *(Last name, first name, initial)*

Graves, G. W., and Whinston, A. B.

6 REPORT DATE	7a TOTAL NO. OF PAGES	7b NO. OF REFS
May, 1967	19	11

8a CONTRACT OR GRANT NO. Nonr 233(75) b PROJECT NO. 047-041 c d	9a ORIGINATOR'S REPORT NUMBER(S) Working Paper No. 117 9b OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
10 AVAILABILITY/LIMITATION NOTICES Distribution of this document is unlimited.	[redacted]
11 SUPPLEMENTARY NOTES Also supported by Nonr-4811(00)	12 SPONSORING MILITARY ACTIVITY

13 ABSTRACT

The paper presents an algorithm for solving the zero-one integer programming problem. In contrast to the usual approach, which reduces the problem to finding the optimal values for variables restricted to values of zero and one, we view the problem as finding an optimal map among a class of mappings of one finite set into another. The basic approach is to characterize the statistical structure of the finite class of maps. Our technique consists of trying to identify the optimal feasible map in a class of maps, by introducing random variables as functionals on the class of maps. We derive explicitly the statistical properties of the class of maps associated withe the zero-one integer programming problem. Using the mean and variance-covariance matrix the idea of confidence level enumeration is developed.

DD FORM 1 JAN 64 1473 0101-807-6800

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Enumeration						
Integer Programming						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, *or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.